IN THE CLAIMS

1. (Previously Presented) A computer-implemented method for maintaining computer security, comprising:

providing a database of known good software;

providing a database of unfamiliar software;

opening a file;

identifying the file being opened;

determining, using a central processing unit, whether an entry exists in the database of known good software for the identified file;

determining, using the central processing unit, whether an entry exists in the database of unfamiliar software for the identified file;

moving the entry from the database of unfamiliar software to the database of known good software if it is determined that the entry has been in the database of unfamiliar software for a predetermined period of time; and

performing at least one of allowing and preventing the opening of the file from continuing based on the result of the determination of whether the entry exists in the database of known good software.

2. (Original) The method of claim 1, wherein the file comprises an executable file.

3. (Original) The method of claim 2, wherein the executable file comprises an application.

4. (Original) The method of claim 1, wherein identifying the file being opened comprises determining a unique value of the file, the unique value being a hash value generated according to a hashing algorithm and comparing the unique value to entries in the database of known good software.

5. (Original) The method of claim 4, wherein the performing at least one of allowing and preventing the opening of the file from continuing comprises allowing the file to continue to be opened if it is determined that the determined unique value corresponds to an entry in the database of known good software.

6. (Cancelled)

7. (Previously Presented) The method of claim 1, further comprising providing date stamp information for each entry in the database of unfamiliar software indicating a date on which the entry was first made.

8. (Previously Presented) The method of claim 1, further comprising providing a value for each entry in the database of unfamiliar software indicating a number of times a file corresponding to the entry was opened.

9. (Original) The method of claim 8, wherein the value comprises the number of times an executable in a file has been executed.

10. (Previously Presented) The method of claim 7, further comprising determining an amount of time an entry has been in the database of unfamiliar software by comparing the date stamp information with a current date.

11. (Cancelled)

12. (Previously Presented) The method of claim 1, further comprising adding an entry to the database of unfamiliar software if an entry for the identified file is not found in at least one of the database of known good software and the database of unfamiliar software.

13. (Previously Presented) The method of claim 1, further comprising placing at least one operating system call hook if it is determined that an entry exists in the database of unfamiliar software.

14. (Previously Presented) The method of claim 13, wherein the operating system call hook notifies a Trojan notification service that a file corresponds to an entry in the database of unfamiliar software.

15. (Previously Presented) The method of claim 14, wherein the Trojan notification service prompts a user for input regarding whether the operating system call should be passed along.

16. (Original) The method of claim 15, wherein opening of the file is allowed to proceed if the operating system call is passed along.

17. (Previously Presented) A system for maintaining computer security, comprising:

a database of known good software;

a database of unfamiliar software;

one or more central processing units operable to execute:

a system for opening a file;

a system for identifying the file being opened;

a system for determining whether an entry exists in the database of known good software for the identified file;

a system for determining whether an entry exists in the database of unfamiliar software for the identified file;

a system for moving the entry from the database of unfamiliar software to the database of known good software if it is determined that the entry has been in the database of unfamiliar software for a predetermined period of time; and

a system for performing at least one of allowing and preventing the opening of the file from continuing based on the result of the determination of whether the entry exists in the database of known good software.

18. (Original) The system of claim 17, wherein the file comprises an executable file.

19. (Original) The system of claim 18, wherein the executable file comprises an application.

20. (Original) The system of claim 17, wherein the system for identifying the file being opened comprises a system for determining a unique value of the file, the unique value being a hash value generated according to a hashing algorithm and a system for comparing the unique value to entries in the database of known good software.

21. (Original) The system of claim 20, wherein the system for performing at least one of allowing and preventing the opening of the file from continuing comprises a system for allowing the file to continue to be opened if it is determined that the determined unique value corresponds to an entry in the database of known good software.

22. (Cancelled)

23. (Previously Presented) The system of claim 17, further comprising a system for providing date stamp information for each entry in the database of unfamiliar software indicating a date on which the entry was first made.

24. (Previously Presented) The system of claim 17, further comprising a system for providing a value for each entry in the database of unfamiliar software indicating a number of times a file corresponding to the entry was opened.

25. (Original) The system of claim 24, wherein the value comprises the number of times an executable in a file has been executed.

26. (Previously Presented) The system of claim 23, further comprising a system for determining an amount of time an entry has been in the database of unfamiliar software by comparing the date stamp information with a current date.

27. (Cancelled)

28. (Previously Presented) The system of claim 17, further comprising a system for adding an entry to the database of unfamiliar software if an entry for the identified file is not found in at least one of the database of known good software and the database of unfamiliar software.

29. (Previously Presented) The system of claim 17, further comprising a system for placing at least one operating system call hook if it is determined that an entry exists in the database of unfamiliar software.

30. (Previously Presented) The system of claim 29, wherein the operating system call hook notifies a Trojan notification service that a file corresponds to an entry in the database of unfamiliar software.

31. (Previously Presented) The system of claim 30, wherein the Trojan notification service prompts a user for input regarding whether the operating system call should be passed along.

32. (Original) The system of claim 31, wherein opening of the file is allowed to proceed if the operating system call is passed along.

33. (Previously Presented) A tangible program storage device including computer executable code for maintaining computer security, comprising:

code for providing a database of known good software;

code for providing a database of unfamiliar software;

code for opening a file;

code for identifying the file being opened;

code for determining whether an entry exists in the database of known good software for the identified file;

code for determining whether an entry exists in the database of unfamiliar software for the identified file;

code for moving the entry from the database of unfamiliar software to the database of known good software if it is determined that the entry has been in the database of unfamiliar software for a predetermined period of time; and

code for performing at least one of allowing and preventing the opening of the file from continuing based on the result of the determination of whether the entry exists in the database of known good software.

34. (Previously Presented) The program storage device of claim 33, wherein the file comprises an executable file.

35. (Previously Presented) The program storage device of claim 34, wherein the executable file comprises an application.

36. (Previously Presented) The program storage device of claim 33, wherein the code for identifying the file being opened comprises code for determining a unique value of the file, the unique value being a hash value generated according to a hashing algorithm and code for comparing the unique value to entries in the database of known good software.

37. (Previously Presented) The program storage device of claim 36, wherein the code for performing at least one of allowing and preventing the opening of the file from continuing comprises code for allowing the file to continue to be opened if it is determined that the determined unique value corresponds to an entry in the database of known good software.

38. (Cancelled)

39. (Previously Presented) The program storage device of claim 33, further comprising code for providing date stamp information for each entry in the database of unfamiliar software indicating a date on which the entry was first made.

40. (Previously Presented) The program storage device of claim 33, further comprising code for providing a value for each entry in the database of unfamiliar software indicating a number of times a file corresponding to the entry was opened.

41. (Previously Presented) The program storage device of claim 40, wherein the value comprises the number of times an executable in a file has been executed.

42. (Previously Presented) The program storage device of claim 39, further comprising code for determining an amount of time an entry has been in the database of unfamiliar software by comparing the date stamp information with a current date.

43. (Cancelled)

44. (Previously Presented) The program storage device of claim 33, further comprising code for adding an entry to the database of unfamiliar software if an entry for the identified file is not found in at least one of the database of known good software and the database of unfamiliar software.

45. (Previously Presented) The program storage device of claim 33, further comprising code for placing at least one operating system call hook if it is determined that an entry exists in the database of unfamiliar software.

46. (Previously Presented) The program storage device of claim 45, wherein the operating system call hook notifies a Trojan notification service that a file corresponds to an entry in the database of unfamiliar software.

47. (Previously Presented) The program storage device of claim 46, wherein the Trojan notification service prompts a user for input regarding whether the operating system call should be passed along.

48. (Previously Presented) The program storage device of claim 47, wherein opening of the file is allowed to proceed if the operating system call is passed along.

49. (Previously Presented) The method of claim 1, wherein a sufficient period of time comprises a month or longer.

50. (Previously Presented) The method of claim 8, further comprising moving the entry from the database of unfamiliar software to the database of known good software if the number of times the file corresponding to the entry was opened is greater than a baseline value.

51. (Previously Presented) The system of claim 17, wherein a sufficient period of time comprises a month or longer.

52. (Previously Presented) The system of claim 24, further comprising a system for moving the entry from the database of unfamiliar software to the database of known good software if the number of times the file corresponding to the entry was opened is greater than a baseline value.

53. (Previously Presented) The system of claim 17, further comprising a processor.

54. (Previously Presented) The program storage device of claim 33, wherein a sufficient period of time comprises a month or longer.

55. (Previously Presented) The program storage device of claim 40, further comprising code for moving the entry from the database of unfamiliar software to the database of known good software if the number of times the file corresponding to the entry was opened is greater than a baseline value.

56. (Currently Amended) A computer-implemented method for computer security, comprising:

identifying a file;

determining, using a central processing unit, whether an entry for the file exists in database of unfamiliar software;

determining, using the central processing unit, quantitative information regarding the file for use in identifying whether the file should be added to a database of known good software, the quantitative information selected from the group consisting of a length of time the entry has been in the database of unfamiliar software, a number of times the file has been opened, and a number of times an executable in the file has been executed;

adding the entry for the file to ~~a~~the database of known good software if the quantitative information exceeds a predetermined value; and

allowing the opening of the file to continue if the database of known good software includes the entry for the file.

57. (Previously Presented) The method of claim 56, further comprising removing the entry for the file from the database of unfamiliar software if the quantitative information exceeds a predetermined value.

58. (Previously Presented) The method of claim 56, further comprising preventing the opening of the file to continue if:

the database of known good software does not include the entry for the file; and

the file attempts a suspicious activity.

59. (Previously Presented) The method of claim 58, wherein a suspicious activity comprises updating a registry.

60. (Previously Presented) The method of claim 58, wherein a suspicious activity comprises opening a second file.